AIL 722: Reinforcement Learning

Lecture 14: Model-Free Policy Evaluation

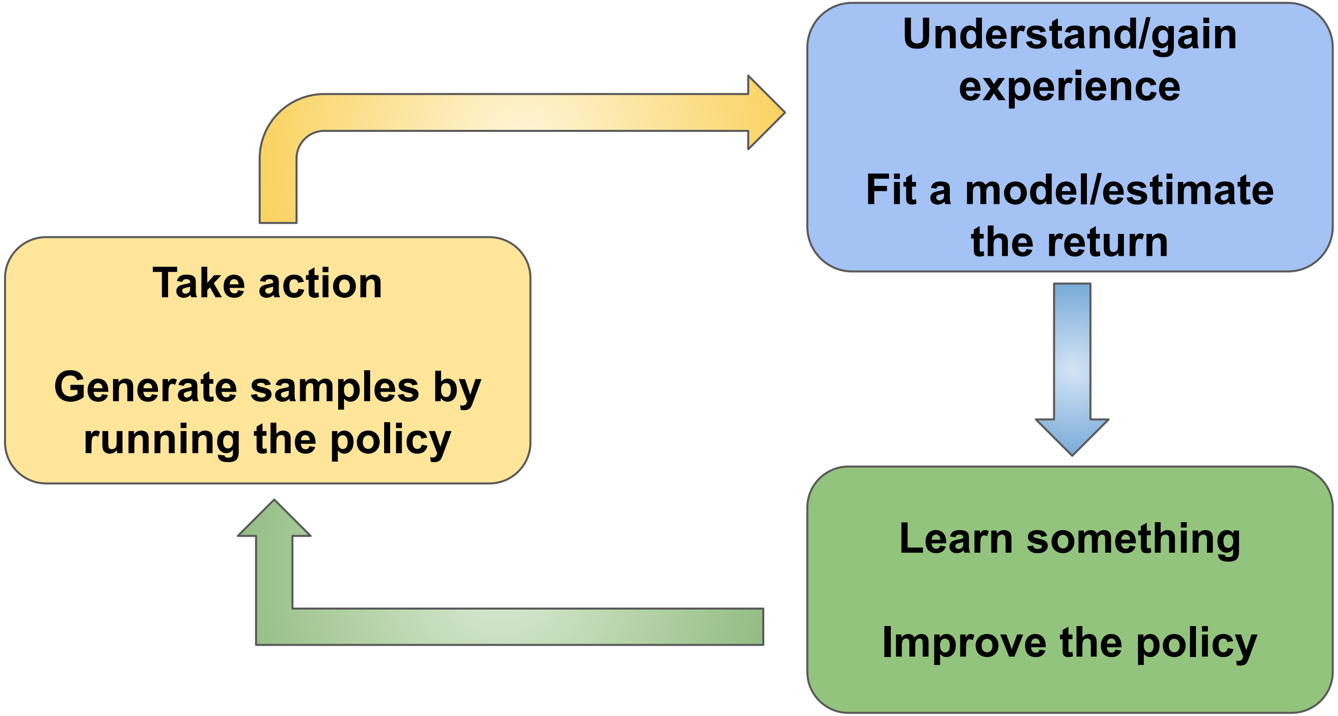
Raunak Bhattacharyya

YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Outline

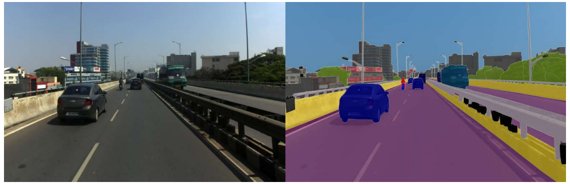
- Model-free policy evaluation
- Monte Carlo
- Temporal Difference

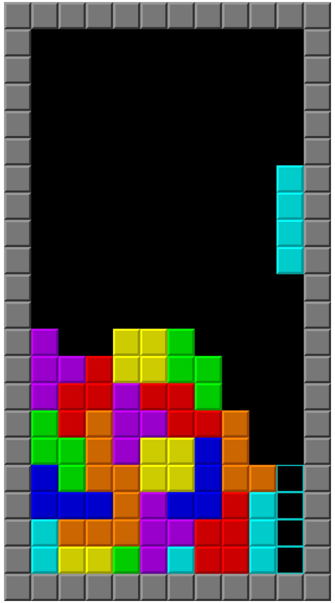
Unifying Anatomy of RL Algorithms



Adapted from Berkeley CS 285, Instructor: Sergey Levine

Towards Real World Problems

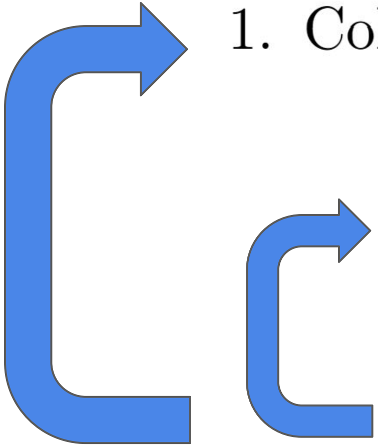




- State:
  - Board configuration
  - Shape of block (tetromino)
- Board is 10x20. And every square could be filled/not filled
- Action: Placement
- Reward: Number of rows eliminated
- Dynamics:
  - Wall change
  - Random next tetromino

How do we use fitted VI?

Fitted QI



1. Collect dataset $\{(s_i, a_i, r_i, s'_i)\}$ using some policy

2. Set $y_i \longleftarrow r(s_i, a_i) + \gamma . \max_{a'_i} Q_\phi(s'_i, a'_i)$

3. Set $\phi \longleftarrow \arg\min_\phi \sum_i \frac{1}{2} \|Q_\phi(s_i, a_i) - y_i\|^2$

How Model Free? Fitted VI vs QI

1. Collect dataset $\{(s_i, a_i, r_i, s'_i)\}$ using some policy

Set $y_i \longleftarrow \max_a \left(r(s_i, a_i) + \gamma . \mathbb{E}\left[V_\phi(s'_i) \right] \right)$

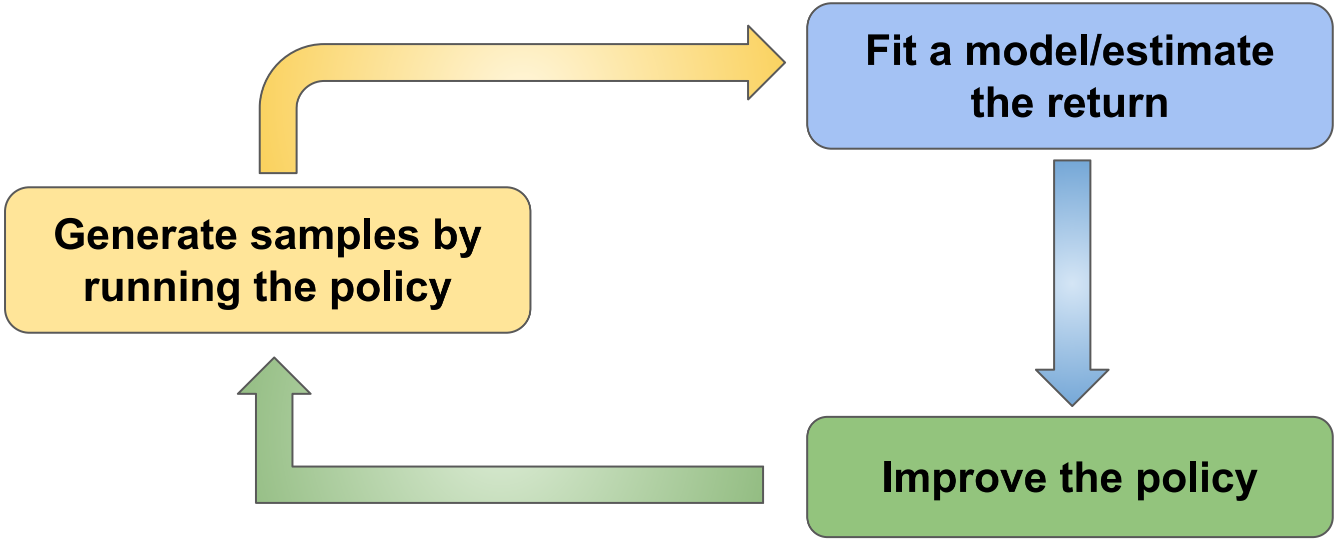
Set $y_i \longleftarrow r(s_i, a_i) + \gamma . \max_{a'_i} Q_\phi(s'_i, a'_i)$

Set $\phi \longleftarrow \arg\min_\phi \sum_i \frac{1}{2} \left\| V_\phi(s_i) - y_i \right\|^2$

Set $\phi \longleftarrow \arg\min_\phi \sum_i \frac{1}{2} \left\| Q_\phi(s_i, a_i) - y_i \right\|^2$

Anatomy of Fitted Q-Iteration

$$Q(s,a) \longleftarrow r(s,a) + \gamma . \max_{a'} Q(s',a')$$



Let's Resurface

- Policy Iteration
- Value Iteration
- Fitted Value Iteration
- Fitted Q Iteration

$$\text{MDP} : \text{Tuple}\langle \mathcal{S}, \mathcal{A}, T, R, \rho \rangle$$

$\mathcal{S}$: State Space

$\mathcal{A}$: Action Space

ρ : Initial State Distribution

$R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$: Reward Function

~~$T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$: Probabilistic Transition Function~~

Model-Free RL

Settings: Motivate Model-Free Approaches

- MDP model is unknown (no transition dyn) but we can sample from it

Autonomous vehicle in traffic

Robotic navigation in unknown envs

Advertising with unknown user behavior

- MDP model is known, but it's easier to sample

Climate models

Robotic navigation

Game playing

Model-Free RL

What was the first phase in policy iteration?

$$V^{\pi}(s^j) = \mathbb{E}_{p_\theta(\tau)}\left[\sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(s_{t'}, a_{t'}) \middle| s_t = s^j\right]$$

How do we do policy evaluation without a model?

Policy Evaluation

$$V^\pi(s_t) = \mathbb{E}\left[\sum_{t'=t}^{T} r(s_{t'}, a_{t'}) \middle| s_t\right]$$

$$V^\pi(s_t) = \mathbb{E}\left[r(s_t, a_t) + \sum_{t'=t+1}^{T} r(s_{t'}, a_{t'}) \middle| s_t\right]$$

$$V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}\left[r(s_t, a_t)\right] + \mathbb{E}\left[\sum_{t'=t+1}^{T} r(s_{t'}, a_{t'})\right]$$

$$V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}\left[r(s_t, a_t)\right] + \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)}\left[V^\pi(s_{t+1})\right]$$

The Bellman equation

Policy Evaluation in Model-Free Setting

$$V^\pi(s) = r\Big(s, \pi(s)\Big) + \gamma \cdot \mathbb{E}_{p(s'|s,\pi(s))}\Big[V^\pi(s')\Big]$$

- Note: S&B calls evaluation as **prediction**

- Note: S&B calls approximating optimal policies as **control**

$$V^\pi(s^j) = \mathbb{E}_{p_\theta(\tau)}\left[\sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(s_{t'}, a_{t'}) \middle| s_t = s^j\right]$$

Monte Carlo Estimation

$$V^{\pi}(s^j) = \mathbb{E}_{p_\theta(\tau)}\left[\sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(s_{t'}, a_{t'}) \middle| s_t = s^j\right]$$

$$\mathbb{E}\big[f(x)\big]$$

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

where $x_1, x_2, \ldots, x_N$ are i.i.d. random samples drawn from the distribution over x.

Monte Carlo Policy Evaluation

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

$$V^{\pi}(s^j) = \mathbb{E}_{p_\theta(\tau)} \left[\sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(s_{t'}, a_{t'}) \middle| s_t = s^j \right]$$

Our goal: $\hat{V}^{\pi}(s^j)$

The **return** is defined as: $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(s_{t'}, a_{t'})$

$$V^{\pi}(s^j) = \mathbb{E}_{p_\theta(\tau)} \left[G_t \middle| s_t = s^j \right]$$

How do we use Monte Carlo estimation to do policy evaluation?

Monte Carlo Policy Evaluation

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$\text{Our goal: } \hat{V}^{\pi}(s^j)$$

- Sample trajectories
- Store the obtained cumulative discounted reward
- Average

Monte Carlo Policy Evaluation

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

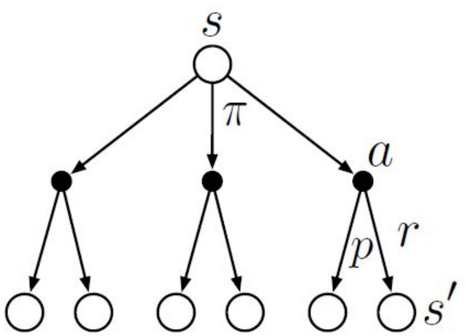
Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Section 5.1, Reinforcement Learning: An Introduction, Sutton & Barto

About Monte Carlo



- Estimates for states are independent
- Estimate for one state does not build upon the estimate of any other state (this was the case in DP)
- Monte Carlo methods do not bootstrap

Computational expense of estimating the value of a single state is independent of the number of states

Section 5.1, Reinforcement Learning: An Introduction, Sutton & Barto